View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Crossref

Hindawi Publishing Corporation
International Journal of Antennas and Propagation
Volume 2014, Article ID 276754, 8 pages
http://dx.doi.org/10.1155/2014/276754

Research Article

Antenna Array Synthesis and Failure Correction Using Differential Search Algorithm

Kerim Guney,[1] Ali Durmus,[2] and Suad Basbug[3]

[1] *Faculty of Engineering, Nuh Naci Yazgan University, 38040 Kayseri, Turkey*
[2] *Department of Electricity and Energy, Vocational College, Erciyes University, 38039 Kayseri, Turkey*
[3] *Department of Computer Technologies, Vocational College, Nevsehir University, 50300 Nevsehir, Turkey*

Correspondence should be addressed to Ali Durmus; alidurmus@gmail.com

Received 24 October 2013; Accepted 31 January 2014; Published 6 March 2014

Academic Editor: Ahmed A. Kishk

Copyright © 2014 Kerim Guney et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Differential search (DS) optimization algorithm is proposed for the synthesis of three different types of linear antenna array design examples. The first group of examples is that DS algorithm is used to locate wide nulls on the linear antenna array patterns by controlling amplitude-only. In these examples, sidelobe levels disposed to rise are also suppressed by using DS algorithm in the same optimization process. In the second group of examples, individual nulls are placed with the help of DS algorithm by controlling the amplitude-only, phase-only, and position-only. The last example is a linear antenna array failure correction example. In order to tolerate the element failures, DS is employed to recalculate the amplitude values of the remaining intact elements of the antenna array. The results show that DS is very capable to solve the linear antenna array optimization problems which have different characteristics.

1. Introduction

Antenna arrays find usage in a large number of application areas, particularly in communication and radar systems. One of the well-known advantages of antenna arrays is that they can be synthesized in a very flexible manner [1]. The main target of antenna array synthesis is to achieve desired radiation pattern by controlling amplitudes, phases, and positions of the antenna array elements. The elimination of unwanted signals which can cause interference is probably the most important ability of antenna array synthesis techniques. There are different kinds of studies on the interference cancellation. In some cases, the arrival direction of interference signal is not exactly known or its location is not permanent and may be slowly changed in time. The wide nulls on the sidelobe region of the array pattern can be required in these kinds of circumstances. If the interference direction is known, an individual null is placed at this certain direction to cancel interference and consequently to increase signal-to-noise ratio. By using this last method, it is possible to achieve very deep null levels. Nulling operations can be performed by controlling amplitude-only [2–14], phase-only [12, 13, 15–21], and position-only [12, 13, 22–28]. The synthesis techniques are not only used for the antenna arrays in good condition but also for the antenna arrays which have failed elements. In some cases, the perturbation on the pattern caused by failed antenna array elements can be tolerated by recalculating the other intact array elements [29–32].

Both deterministic and stochastic optimization algorithms have been widely used for the synthesis of antenna arrays. However, the popularity of the stochastic algorithms has been increased since they are more flexible than their counterparts. Additionally, the stochastic algorithms are very skillful to escape from local minima. Several versions of stochastic algorithms are utilized to synthesize antenna arrays in the literature. Genetic algorithm (GA) [3, 16, 17, 30, 31], differential evolution (DE) algorithm [6, 27], particle swarm optimization (PSO) algorithm [14, 23], tabu search algorithm [4, 18], touring ant colony algorithm [5, 22], clonal selection algorithm [7, 19, 24, 26], bees algorithm [8, 21, 25], bacterial foraging algorithm (BFA) [9, 20, 28], immune algorithm [10], plant growth simulation algorithm [11], seeker optimization algorithm [12], harmony search algorithm [13], and firefly algorithm (FA) [32] can be given as the examples of these stochastic optimization algorithms.

In this paper, differential search algorithm (DS) [33] is employed to solve three different types of antenna array optimization problems. The first is the wide nulling on the array pattern by controlling amplitude-only. The second group of optimization problem is the single nulling on the array pattern by controlling amplitude-only, phase-only, and position-only. The failure correction is the last kind of optimization problem presented in this paper. DS originally proposed in [33] is a stochastic search optimization algorithm that mimics the migration behaviors of organisms which use the Brownian-like random-walk movement. In DS algorithm, the population members are represented by artificial organisms. The artificial organisms change their positions in the solution space by so-called migration movement. The population members stay on their new positions temporarily in an iteration cycle. They make their decision about whether or not to stay on the new position in regard to the cost function. This movement goes on iteratively until the stopping criteria are satisfied. In [33], the DS algorithm was used to solve the problem of transforming the geocentric cartesian coordinates into geodetic coordinates and its performance was compared with the performances of eight computational intelligence algorithms which consist of artificial bee colony, self-adaptive DE, adaptive DE, strategy adaptation based DE, DE with ensemble of parameters, gravitational search algorithm, PSO, and covariance matrix adaptation evolution strategy, through use of a set of test data containing 100,000 test points. The statistical tests realized for the comparison of performances indicated that the problem-solving success of DS algorithm in transforming the geocentric Cartesian coordinates into geodetic coordinates is higher than that of computational intelligence algorithms. In [34], DS was used to solve two famous economic dispatch problems with practical constraints for 6-unit and 15-unit systems. The results showed that DS can achieve more economical solutions than PSO, self-organizing hierarchical PSO, GA, BFA, FA, and modified FA.

2. Problem Formulation

The array factor of a linear antenna array in azimuth plane (x-y plane) with M elements placed along the x-axis is given by

$$\mathrm{AF}(\theta) = \sum_{m=1}^{M} I_m e^{j(kd_m \sin\theta + \delta_m)}, \tag{1}$$

where θ is the scanning angle from broadside, k is the wavenumber ($k = 2\pi/\lambda$), and I_m, d_m, and δ_m are the amplitude, location, and phase of the mth element, respectively. The array elements are considered as isotropic sources.

The array factor can be written as the following format in case that the antenna array elements are symmetrically located and excited around the center of the linear array:

$$\mathrm{AF}(\theta) = 2\sum_{m=1}^{H} I_m \cos\left(kd_m \cos\theta + \delta_m\right), \tag{2}$$

where the number of array elements is even and $H = M/2$.

The first cost function used for the wide nulling and failure correction examples in this paper is formulated as

$$C_1 = C_{\mathrm{MSL}} + C_{\mathrm{FNBW}} + C_{\mathrm{NULL}}, \tag{3}$$

where C_{MSL} is the function used to suppress the maximum side lobe level (MSL) values; C_{FNBW} function makes sure that the obtained the first null beam width (FNBW) value is confined in the range of determined maximum FNBW values; C_{NULL} is used to locate single or wide nulls on sidelobes. The function C_{MSL} is given as

$$C_{\mathrm{MSL}} = \int_{-\pi/2}^{\theta_{\mathrm{null1}}} \chi_{\mathrm{MSL}}(\theta)\, d\theta + \int_{\theta_{\mathrm{null2}}}^{\pi/2} \chi_{\mathrm{MSL}}(\theta)\, d\theta, \tag{4}$$

where θ_{null1} is the first null on the left side of the main beam and θ_{null2} is the first null on the right side of the main beam. $\chi_{\mathrm{MSL}}(\theta)$ function is used to obtain the values which exceed the desired MSL (MSL_d) in the sidelobe region and it can be calculated by the following expression:

$$\chi_{\mathrm{MSL}}(\theta) = \begin{cases} \left(20\log\left(\mathrm{AF}(\theta)_{\mathrm{normalized}}\right) - \mathrm{MSL}_d\right)^2, \\ \qquad \text{for } 20\log\left(\mathrm{AF}(\theta)_{\mathrm{normalized}}\right) > \mathrm{MSL}_d \\ 0, \\ \qquad \text{elsewhere.} \end{cases} \tag{5}$$

C_{FNBW} function in (3) is given by

$$C_{\mathrm{FNBW}} = \begin{cases} \left(\mathrm{FNBW}_o - C_{\mathrm{fnbwmax}}\right)^2, & \text{for } \mathrm{FNBW}_o > C_{\mathrm{fnbwmax}} \\ 0, & \text{elsewhere,} \end{cases} \tag{6}$$

where FNBW_o is the obtained FNBW value by DS and C_{fnbwmax} is the desired maximum FNBW.

In this paper two different nulling methods are presented: wide nulls and single nulls. In order to produce wide nulls, C_{NULL} in (3) can be written as follows:

$$C_{\mathrm{NULL}}(\theta) = \begin{cases} \left(20\log\left(\mathrm{AF}(\theta)_{\mathrm{normalized}}\right) - \mathrm{NULL}_d\right)^2, \\ \qquad \text{for } 20\log\left(\mathrm{AF}(\theta)_{\mathrm{normalized}}\right) > \mathrm{NULL}_d \\ 0, \\ \qquad \text{elsewhere,} \end{cases} \tag{7}$$

where NULL_d is the null depth level (NDL) which is desired at the predetermined angle. To produce wide nulls, C_{NULL} function is used for the every angle in the range of the wide null region and the total values achieved by C_{NULL} are added to the cost function.

```
Set the initial random positions for the super_organism members
Evaluate the cost functions for each members.
while (stopping criterion is not satisfied)
{
    Select the donor members by a random shuffling mechanism.
    Produce the scale value by using a gamma random number generator (10)
    Determine the Stopover Sites by using (11)
    Determine the population members to be used for the stop over process.
    Set new random positions in the solution space for the members which exceed the boundaries.
}
Display the best result.
```

Algorithm 1: Pseudocode of DS algorithm.

In this paper, the following second cost function will be used for the single nulls produced by controlling amplitude-only, phase-only, and position-only:

$$C_2 = \sum_{\theta=-90^\circ}^{90^\circ} \left[W(\theta) \left| \mathrm{AF}_o(\theta) - \mathrm{AF}_d(\theta) \right| + \mathrm{ESL}(\theta) \right], \quad (8)$$

where $\mathrm{AF}_o(\theta)$ function produces the values obtained by using DS and $\mathrm{AF}_d(\theta)$ gives the desired array pattern values. $W(\theta)$ and $\mathrm{ESL}(\theta)$ are employed to control the NDL and MSL, respectively.

3. Differential Search Algorithm

Superorganism is a collection of single creatures that together possess the functional organization implicit in the formal definition of organism [33, 35]. The superorganism having many individuals migrates to fruitful lands when they need much nutrition, usually on a seasonal basis. Some migration movements can be explained by Brownian-like random-walk. Superorganism continuously checks new locations to find more fertile areas. If a more fertile location is detected, the superorganism moves itself to this location.

In DS algorithm [33], a superorganism is a population which consists of solution members distributed randomly to the solution space. This artificial superorganism tries to find the global minimum value in the solution space by performing migration movement. The superorganism is always in search of new locations which includes better solutions (much food) during the migration process. Searching activity mainly depends on a stochastic mechanism. The individual of the superorganism which finds better location than the previous one moves to this new location temporarily until the global minimum is found.

Artificial organisms as the members of the superorganism are denoted by X_i, $i = \{1, 2, 3, \ldots, P\}$, where P is the number of population. The iteration index is represented by $g = 1, 2, 3, \ldots, G$, where G is the maximum iteration number. Assuming that D is the size of the problem, every individual of the superorganism can be described as $x_{i,j}$, $j = \{1, 2, 3, \ldots, D\}$.

The initial random values of the organisms are given as the following formulation:

$$x_{i,j} = \mathrm{rand} \cdot \left(\mathrm{up}_j - \mathrm{low}_j \right) + \mathrm{low}_j, \quad (9)$$

where up_j and low_j are upper and lower values of the solutions, respectively. In this way, the initial values are always in the range of permitted values. rand is a random number distributed uniformly.

In order to describe DS algorithm, one more term should also be defined: stopover sites. The locations where the organisms stay until finding better places are named as stopover sites. A donor is randomly selected from the population to discover new stopover sites. A scale value is needed to control the step size of superorganism. The scale value is produced as the following:

$$\mathrm{Scale} = \mathrm{randg} \cdot \left[2 \cdot \mathrm{rand}_1 \right] \cdot \left(\mathrm{rand}_2 - \mathrm{rand}_3 \right), \quad (10)$$

where randg is a gamma random number generator. rand_1, rand_2, and rand_3 functions generate uniform random numbers in the range of $[0, 1]$. After producing the scale value, the new stopover site is calculated as follows:

$$\begin{aligned} \mathrm{StopoverSite} = {} & \mathrm{Superorganism} + \mathrm{Scale} \\ & \times \left(\mathrm{donor} - \mathrm{Superorganism} \right). \end{aligned} \quad (11)$$

If the new stopover is out of the range, a random location is assigned as stopover location instead of the calculated value. If the new stopover site is better than the current location, superorganism moves to the new stopover site. Therefore, the members of the superorganism perform the searching activity until the global minimum is found. A pseudocode of the DS algorithm is shown in Algorithm 1. The detailed information about DS algorithm can be found in [33].

4. Numerical Results

In this paper, in order to test the capabilities of DS algorithm, three different kinds of examples are given. The first group of examples is to place wide nulls on the array pattern by controlling amplitude-only. For the second group of

Table 1: FNBW, MSL, NDL, and DRR values obtained by DS and CRPSO [14] for the first example (12 elements).

	FNBW (deg.)	MSL (dB)	NDL (dB)	DRR
DS	36.50	−18.37	−45.33	5.50
CRPSO [14]	38.84	−17.22	−41.78	5.52

Table 2: FNBW, MSL, NDL, and DRR values obtained by DS and CRPSO [14] for the second example (16 elements).

	FNBW (deg.)	MSL (dB)	NDL (dB)	DRR
DS	29.50	−20.96	−53.30	15.17
CRPSO [14]	30.60	−17.49	−53.04	Infinitive

Table 3: FNBW, MSL, NDL, and DRR values obtained by DS and CRPSO [14] for the third example (20 elements).

	FNBW (deg.)	MSL (dB)	NDL (dB)	DRR
DS	22.50	−19.65	−47.75	8.24
CRPSO [14]	23.02	−17.61	−45.56	57.88

examples, DS is used to achieve array patterns with individual nulls imposed at the directions of interference by controlling amplitude-only, phase-only, and position-only. In the last example of this paper, DS is implemented as a failure correction technique. A personal computer having i7 processor running at 1.73 GHz and 4 GB RAM is employed for all simulation examples. The software used for the simulations is MATLAB on Windows 7 operating system. For all examples in this paper, the number of population size and the number of iteration are 100 and 2100, respectively.

4.1. Wide Nulls. DS is used for the synthesis of antenna array patterns with wide nulls by controlling amplitude-only values of the array elements. Three different linear antenna arrays which have 12, 16, and 20 elements with $\lambda/2$ interelement spacing are considered for these three examples. The optimization processes take approximately 125, 165, and 207 seconds for the linear antenna arrays with 12, 16, and 20 elements, respectively. The function in (3) is used as cost function in the simulations.

The antenna array used in the first example simulations has 12 equispaced elements. DS is used to calculate the amplitude values of array elements in order to achieve an array pattern having wide nulls in the range of [46°, 71°] and [108°, 134°]. The array pattern obtained by DS is compared with the pattern of craziness PSO (CRPSO) [14] in Figure 1. FNBW, MSL, NDL, and dynamic range ratio (DRR = $|I_{\max}/I_{\min}|$) values obtained by DS and CRPSO [14] are tabulated in Table 1. In Figure 1 and Table 1, it can be seen that FNBW, MSL, and NDL values achieved by DS are better than those of CRPSO [14], whereas DRR values are about the same for both algorithms.

In the second example, an antenna array with 16 equispaced elements is considered. The targeted null ranges of the array pattern are in [60°, 75°] and [104°, 120°]. Figure 2 shows the patterns obtained by DS and CRPSO [14]. FNBW, MSL, NDL, and DRR values achieved by DS and CRPSO [14] are shown in Table 2. In Figure 2 and Table 2, it is very clear that FNBW, MSL, and DRR values achieved by DS are better than those of CRPSO [14], whereas NDL values calculated by both DS and CRPSO [14] are about the same.

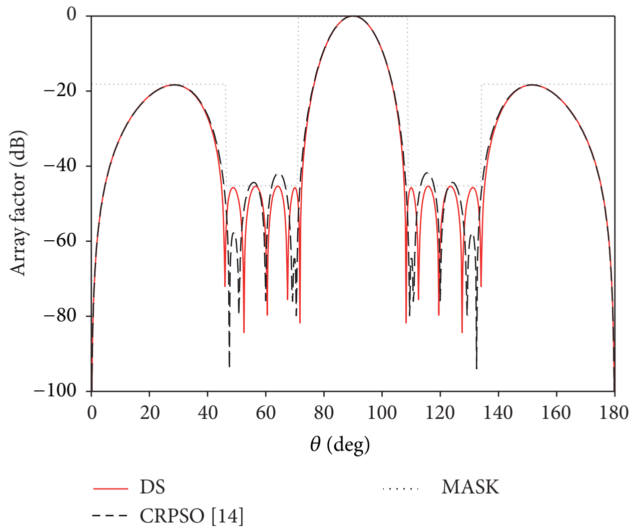


Figure 1: The radiation patterns with wide nulls at [46°, 71°] and [108°, 134°] obtained by DS and CRPSO [14] for the first example (12 elements).

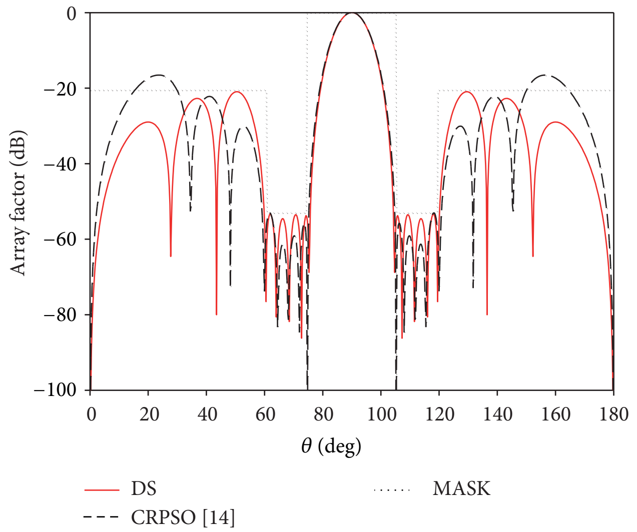


Figure 2: The radiation patterns with wide nulls at [60°, 75°] and [104°, 120°] obtained by DS and CRPSO [14] for the second example (16 elements).

The linear antenna array utilized in the simulations has 20 equispaced elements for the third example. The desired wide null ranges were determined as in [66°, 78°] and [101°, 113°]. The patterns obtained by using DS and CRPSO [14] are illustrated in Figure 3. FNBW, MSL, NDL, and DRR values calculated by DS and CRPSO [14] are shown in Table 3. In Figure 3 and Table 3, it can be said that FNBW, MSL, NDL, and DRR values achieved by DS are better than those of CRPSO [14].

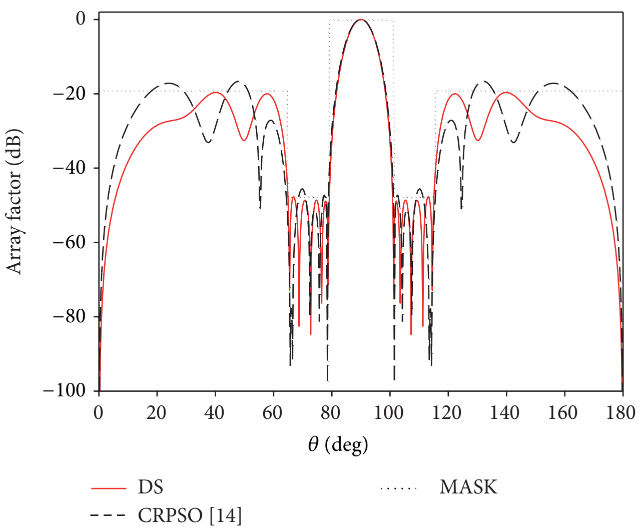


Figure 3: The radiation patterns with wide nulls at [66°, 78°] and [101°, 113°] obtained by DS and CRPSO [14] for the third example (20 elements).

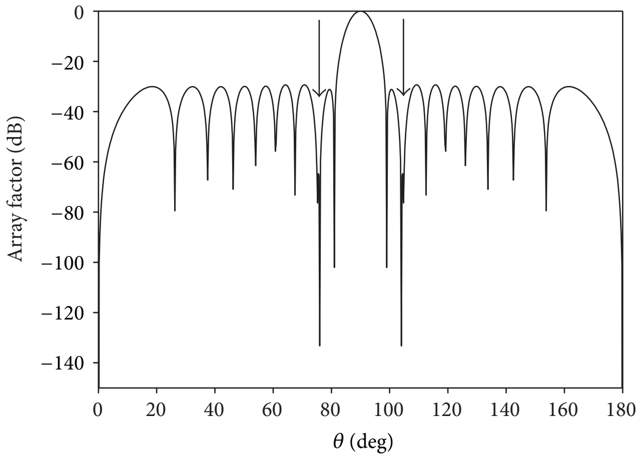


Figure 4: Radiation pattern obtained by amplitude-only control with one imposed null at 104°.

Table 4: The amplitude values calculated by DS for the antenna arrays with 12, 16, and 20 elements.

Arrays	Amplitude values
12 elements (first example)	1.0000, 0.9996, 0.6034, 0.6055, 0.1816, and 0.1891
16 elements (second example)	0.9745, 0.9999, 0.8518, 0.5440, 0.5587, 0.3664, 0.0659, and 0.1960
20 elements (third example)	0.6604, 0.9929, 0.5991, 0.6066, 0.4668, 0.4443, 0.3396, 0.1298, 0.1205, and 0.1370

The amplitude values calculated by DS in the first, second, and third examples are tabulated in Table 4. These examples clearly show the capacity of DS to synthesize the array pattern with wide nulls imposed at the direction of interference.

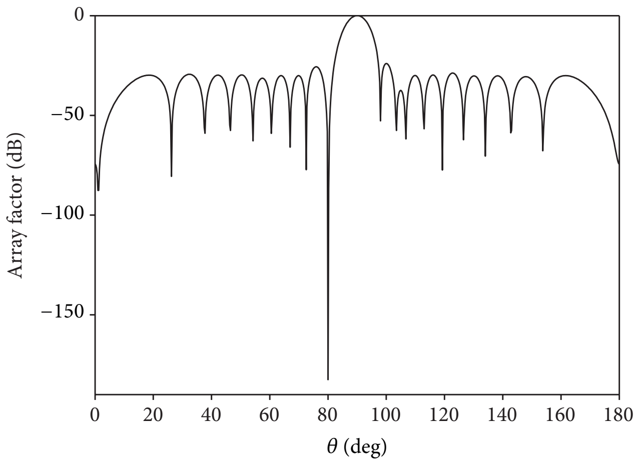


Figure 5: Radiation pattern obtained by phase-only control with one imposed null at 80°.

4.2. Single Null. DS is used to synthesize three different antenna array patterns with single nulls by controlling the amplitude-only, phase-only, and position-only of the array elements. For this aim, a 30 dB Chebyshev pattern of an antenna array having 20 equispaced isotropic elements with $\lambda/2$ interelement spacing is used as initial pattern for the simulations. The time consumption for all optimization processes of the single null examples is about 202 seconds. The cost function in (8) is employed for these three examples.

In the fourth example, the pattern nulling is achieved by controlling only the element amplitudes. The nulling direction θ_i is chosen as 104°. The cost function parameters given in (8) are defined as follows:

$$\mathrm{AF}_d(\theta) = \begin{cases} 0, & \text{for } \theta = \theta_i \\ \text{Initial\_pattern}, & \text{elsewhere}, \end{cases}$$
$$W(\theta) = \begin{cases} 200, & \text{for } \theta = \theta_i \\ 1, & \text{elsewhere}, \end{cases} \tag{12}$$
$$\mathrm{ESL}(\theta) = \begin{cases} 50, & \text{if MSL} > -29\,\text{dB} \\ 0, & \text{elsewhere}. \end{cases}$$

The pattern with null at 104° obtained by DS is shown in Figure 4. Due to the symmetrical structure of the array, another null with the same NDL can also be observed at 76°. The NDL and MSL values of the pattern in Figure 4 are −133 dB and −29.33 dB, respectively.

DS is utilized to obtain an optimized pattern with a single null at $\theta_i = 80°$ by controlling the phase-only in the fifth example. MSL is restricted with −24 dB. The other cost function definition parameters are determined as in the fourth example. Figure 5 shows the pattern achieved by using DS. It can be seen from Figure 5 that there is a single null on the pattern. The NDL value of this null is −182.5 dB and the MSL value of the pattern is −24.0 dB.

For the sixth example, the element positions of the array are calculated by DS algorithm to locate a null at $\theta_i = 105°$ on the array pattern. MSL value is limited by −28 dB. The pattern

Table 5: Antenna arrays parameters values obtained by DS for the fourth, fifth, and sixth examples.

Parameters	Parameters values
Amplitude (fourth example)	1.00000, 0.98124, 0.94043, 0.86849, 0.75833, 0.61671, 0.46289, 0.32721, 0.22770, and 0.29669
Phase (deg.) (fifth example)	80.62, 162.75, 182.03, 175.35, 113.63, −24.90, −210.79, −165.96, −176.45, and −166.45
Position (λ) (sixth example)	0.24982, 0.76064, 1.25961, 1.74163, 2.21348, 2.67524, 3.14991, 3.65736, 4.20858, and 4.75772

Table 6: MSL, DRR, and FNBW values obtained by obtained by DS and FA [32].

	MSL (dB)	DRR	FNBW (deg.)
DS	−35.02	45.74	19
FA [32]	−34.19	49.75	19

Table 7: Corrected amplitude weights obtained by DS.

	Amplitude values
32 elements (seventh example)	1.0000, 0.9663, 0.8956, 0.8024, 0.6856, 0.5615, 0.4367, 0.3286, 0.2192, 0.1773, 0, 0, 0.0219, 0, 0, and 0

obtained by DS is shown in Figure 6. Due to the symmetry, it is possible to see that there is another null generated at 75°. The NDL and MSL values obtained by DS algorithm are −151.7 dB and −28.11 dB, respectively.

Table 5 presents the amplitude, phase, and position values calculated by DS in the fourth, fifth, and sixth examples, respectively. From the null depth and the maximum side lobe level points of view, the performances of the patterns in these examples are very good. The nulling technique based on DS algorithm preserves the characteristics of the initial Chebyshev pattern with little pattern disturbance except for the nulling directions.

4.3. Failure Correction. For the seventh example, a 35 dB Chebyshev antenna array having 32 equispaced elements with $\lambda/2$ interelement spacing is considered. The function in (3) is used as a cost function in this simulation. It is assumed that 1st, 2nd, 3rd, 5th, 6th, 27th, 28th, 30th, 31th, and 32th elements of the antenna array failed. Under this condition, the pattern of the antenna array is disturbed by raising its MSL value to −21.29 dB. DS algorithm is used to recalculate the values of the remaining intact elements in order to compensate for the element failures. The optimization process of the failure correction example takes about 325 seconds. Figure 7 shows the patterns of the Chebyshev array, failed array, and corrected arrays by DS. It can be seen in Figure 7 that DS succeeded to suppress the MSL value in an acceptable level. In Table 6, The MSL, DRR, and FNBW values of the pattern obtained by using DS are compared with those of the pattern obtained by using FA [32]. It is clear from Figure 7 and Table 6 that MSL and DRR values obtained by DS are better than those of FA whereas their FNBW values are the same. The amplitude values corrected by DS are tabulated in Table 7.

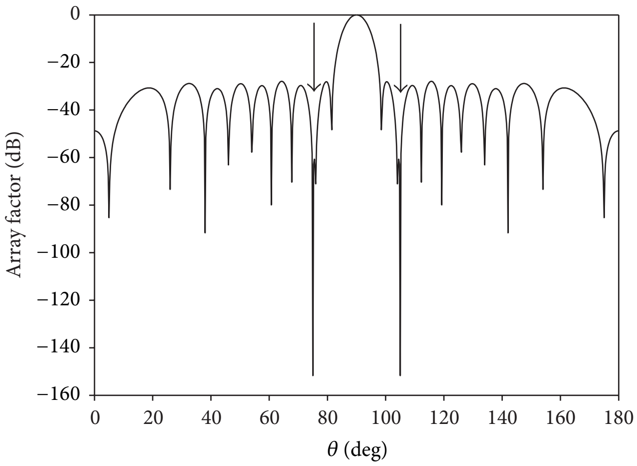


Figure 6: Radiation pattern obtained by position-only control with one imposed null at 105°.

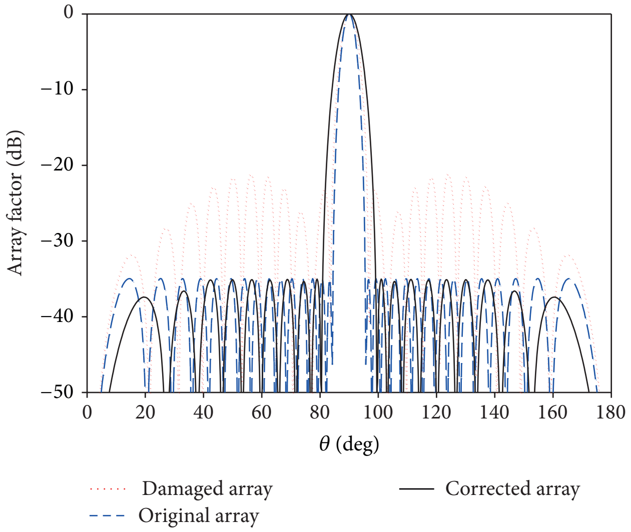


Figure 7: The damaged, original, and corrected radiation patterns.

5. Conclusions

In this paper, DS optimization algorithm is employed to solve three different types of antenna array problems. In the first group of examples, wide nulls on the array patterns are attained by using DS algorithm. In the same nulling concept, for the second group of examples, DS is also utilized in order to achieve single nulls on the array patterns by controlling amplitude-only, phase-only, and position-only. In the last example, DS is used to mitigate the negative effects of the failed elements on the array pattern by optimizing amplitude values of the remaining intact elements. It can be concluded that DS has capability of solving different types of antenna array synthesis problems. DS can be used as an alternative to other antenna array synthesis algorithms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] R. J. Mailloux, *Phased Array Antenna Handbook*, Artech House, Boston, 2005.

[2] H. M. Ibrahim, "Null steering by real-weight control—a method of decoupling the weights," *IEEE Transactions on Antennas and Propagation*, vol. 39, no. 11, pp. 1648–1650, 1991.

[3] F. J. Ares-Pena, J. A. Rodriguez-Gonzalez, E. Villanueva-Lopez, and S. R. Rengarajan, "Genetic algorithms in the design and optimization of antenna array patterns," *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 3, pp. 506–510, 1999.

[4] K. Guney and A. Akdagli, "Null steering of linear antenna arrays using a modified tabu search algorithm," *Progress in Electromagnetics Research*, vol. 33, pp. 167–182, 2001.

[5] N. Karaboga, K. Güney, and A. Akdagli, "Null steering of linear antenna arrays with use of modified touring ant colony optimization algorithm," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 12, no. 4, pp. 375–383, 2002.

[6] S. Yang, Y. B. Gan, and A. Qing, "Antenna-array pattern nulling using a differential evolution algorithm," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 14, no. 1, pp. 57–63, 2004.

[7] B. Babayigit, A. Akdagli, and K. Guney, "A clonal selection algorithm for null synthesizing of linear antenna arrays by amplitude control," *Journal of Electromagnetic Waves and Applications*, vol. 20, no. 8, pp. 1007–1020, 2006.

[8] K. Guney and M. Onay, "Amplitude-only pattern nulling of linear antenna arrays with the use of bees algorithm," *Progress in Electromagnetics Research*, vol. 70, pp. 21–36, 2007.

[9] K. Guney and S. Basbug, "Interference suppression of linear antenna arrays by amplitude-only control using a bacterial foraging algorithm," *Progress in Electromagnetics Research*, vol. 79, pp. 475–497, 2008.

[10] K. Guney and B. Babayigit, "Amplitude-only pattern nulling of linear antenna arrays with the use of an immune algorithm," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 18, no. 5, pp. 397–409, 2008.

[11] K. Guney, A. Durmus, and S. Basbug, "A plant growth simulation algorithm for pattern nulling of linear antenna arrays by amplitude control," *Progress In Electromagnetics Research B*, no. 17, pp. 69–84, 2009.

[12] K. Guney and S. Basbug, "Seeker optimization algorithm for interference suppression of linear antenna arrays by controlling position-only, phase-only, and amplitude-only," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 21, no. 5, pp. 505–518, 2011.

[13] K. Guney and M. Onay, "Optimal synthesis of linear antenna arrays using a harmony search algorithm," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15455–15462, 2011.

[14] D. Mandal, R. Kumar, R. Kar, and S. P. Ghoshal, "Wide nulls control of linear antenna arrays using craziness based particle swarm optimization," in *Proceedings of the IEEE Student Conference on Research and Development (SCOReD '11)*, pp. 189–193, Kuala Lampur, Malaysia, December 2011.

[15] H. Steyskal, R. A. Shore, and R. L. Haupt, "Methods for null control and their effects on the radiation pattern," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 404–409, 1986.

[16] R. L. Haupt, "Phase-only adaptive nulling with a genetic algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 6, pp. 1009–1015, 1997.

[17] W.-P. Liao and F.-L. Chu, "Application of genetic algorithms to phase-only null steering of linear arrays," *Electromagnetics*, vol. 17, no. 2, pp. 171–183, 1997.

[18] A. Akdagli and K. Guney, "Null steering of linear antenna arrays by phase perturbations using modified tabu search algorithm," *Journal of Communications Technology and Electronics*, vol. 49, no. 1, pp. 37–42, 2004.

[19] K. Guney, B. Babayigit, and A. Akdagli, "Interference suppression of linear antenna arrays by phase-only control using a clonal selection algorithm," *Journal of the Franklin Institute*, vol. 345, no. 3, pp. 254–266, 2008.

[20] K. Guney and S. Basbug, "Phase-only pattern nulling of linear antenna arrays with the use of a bacterial foraging algorithm," *Neural Network World*, vol. 18, no. 4, pp. 257–273, 2008.

[21] K. Guney and M. Onay, "Bees algorithm for interference suppression of linear antenna arrays by controlling the phase-only and both the amplitude and phase," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3129–3135, 2010.

[22] A. Akdagli, K. Güney, and D. Karaboga, "Pattern nulling of linear antenna arrays by controlling only the element positions with the use of improved touring ant colony optimization algorithm," *Journal of Electromagnetic Waves and Applications*, vol. 16, no. 10, pp. 1423–1441, 2002.

[23] M. M. Khodier and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 8, pp. 2674–2679, 2005.

[24] K. Guney, B. Babayigit, and A. Akdagli, "Position only pattern nulling of linear antenna array by using a clonal selection algorithm (CLONALG)," *Electrical Engineering*, vol. 90, no. 2, pp. 147–153, 2007.

[25] K. Guney and M. Onay, "Bees algorithm for null synthesizing of linear antenna arrays by controlling only the element positions," *Neural Network World*, vol. 17, no. 2, pp. 153–169, 2007.

[26] B. Babayigit, K. Guney, and A. Akdagli, "A clonal selection algorithm for array pattern nulling by controlling the positions of selected elements," *Progress in Electromagnetics Research B*, vol. 6, pp. 257–266, 2008.

[27] C. Lin, A. Qing, and Q. Feng, "Synthesis of unequally spaced antenna arrays by using differential evolution," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 8, pp. 2553–2561, 2010.

[28] K. Guney and S. Basbug, "Bacterial foraging algorithm for null synthesizing of linear antenna arrays by controlling only the element positions," *Journal of Optimization Theory and Applications*, vol. 150, no. 3, pp. 635–656, 2011.

[29] R. J. Mailloux, "Array failure correction with a digitally beamformed array," *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 12, pp. 1543–1550, 1996.

[30] B.-K. Yeo and Y. Lu, "Array failure correction with a genetic algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 5, pp. 823–828, 1999.

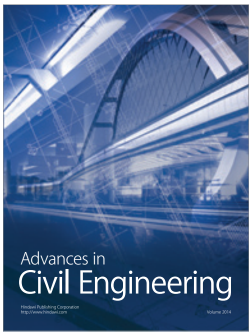
[31] J. A. Rodríguez, F. Ares, E. Moreno, and G. Franceschetti, "Genetic algorithm procedure for linear array failure correction," *Electronics Letters*, vol. 36, no. 3, pp. 196–198, 2000.

[32] N. S. Grewal, M. Rattan, and M. S. Patterh, "A linear antenna array failure correction using firefly algorithm," *Progress in Electromagnetics Research M*, vol. 27, pp. 241–254, 2012.

[33] P. Civicioglu, "Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm," *Computers and Geosciences*, vol. 46, pp. 229–247, 2012.

[34] M. H. Sulaiman and M. W. Mustafa, "Solving economic dispatch problems with practical constraints utilizing differential search algorithm," in *Proceedings of the IEEE Symposium on Humanities, Science and Engineering Research (SHUSER '13)*, pp. 310–314, Pinang, Malaysia, 2013.

[35] D. S. Wilson and E. Sober, "Reviving the superorganism," *Journal of Theoretical Biology*, vol. 136, no. 3, pp. 337–356, 1989.



Hindawi

Submit your manuscripts at
http://www.hindawi.com